ConneXions
The Interoperability Report

May 1990 Volume 4, No. 5

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop, Inc., 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. 415-941-3399. Fax: 415-949-1779.

ISSN 0894-5926

From the Editor

In previous issues of *ConneXions* we have made references to the *X Window System™*. We've reviewed books on the subject, and talked about the cooperative demonstrations of **X** at the INTEROP® exhibition. It is time to discover what **X** really is. To help us understand this popular window system, we turned to Bill Jolitz of TeleMuse.

The OSI and TCP/IP communities have always had very different working styles. A certain amount of mutual sceptisism has contributed to a prevailing "us versus them" attitude. It was encouraging, therefore, when members of *ANSI X3S3.3* (the US representatives to ISO) met jointly with the *Internet Engineering Task Force* (IETF) in 1987. This joint meeting led to a fruitful cooperation which has continued since then. Lyman Chapin describes the areas of mutual interest between the two groups, in an article starting on page 14.

The Telnet LINEMODE option was recently defined and published in RFC 1116. It provides a standard method for moving the bulk of character input processing to the local Telnet application, thus reducing the number and frequency of packets sent over the network, and the user's frustration at having to wait for single characters to be echoed remotely over a slow and/or congested network. Dave Borman from Cray Research describes the features of this option on page 16.

In our series profiling large internets we feature an overview of *CREN—The Corporation for Research and Educational Networking*. CREN was formed from the merger of CSNET and BITNET and is one of the largest academic networks in the world. The network uses a variety of networking technologies, and the merger represents a major technical challenge. The article is by Mark Laubach of H-P who is the chairman of the Technical Committee for CREN.

We are pleased and honored to welcome Lyman Chapin of Data General, and chairman of ANSI's X3S3.3, as the fifth member of the Editorial Advisory Board for *ConneXions* (see back cover). Lyman's background with the OSI suite of protocols adds another facet to this already impressive body of knowledge.

Also, in this our third anniversary issue: A book review, a letter to the Editor, some announcements, and information about a new book on computer viruses.

It is not too late to sign up for our *Internetworking Tutorials* which will be offered in London, England May 29–June 1, and in Dallas, Texas June 11–14. For more information, or to register, call Interop, Inc. at 415-941-3399 or 1-800-INTEROP. (In the United Kingdom call: 0800-891-464 or 01-231-7835).

X Windows: More than Just a Pretty Face

by William Jolitz, TeleMuse

We have all watched the computer processor market evolve over the past ten years from the expensive and limited computation power of older machines to low-cost, high-powered desktop systems. However, as more capability has been placed before the customer, demand for more comprehensive and exotic displays for data and computation has increased. The bitmap display, no longer a research lab curiosity, is fast becoming as standard a computer accessory as a keyboard. Of course, one would have had to have been living in a cave for the last ten years for this to be new!

Motivations

With the development of new display technology and enough computer "horsepower" to drive it, the process by which one interacts and perceives information through computers is altered. The first popular computer to exploit new display technology in a significant way, the Apple Macintosh, utilized the Xerox-developed desktop metaphor. Instead of a keyboard, the Macintosh allowed neophyte users to control the computer via a "point-and-shoot" mouse, thus avoiding the uncomfortable learning curve of an arcane command language.

In more intensive applications, bitmapped window systems allow for more than just simplified pedagogy. Other major motivations beyond the ability to "point-and-shoot" can include:

- The ability to represent data in forms other than text, such as the currently popular graphs, pie charts and and bar charts. This technology is not limited to just these examples, however. Other uses, such as instrument simulation (e.g., flight simulator gauges), Mecca-maps that topologically twist geography to a given perspective, and others yet to be invented will also benefit from this new technology.

- Multitasking bitmapped window systems through the use of multiple windows. Ease of access to many applications tools to solve complex tasks becomes a powerful motivator towards this technology. Indeed, as bitmapped window systems begin to eclipse older display technologies, there will be a migration from the PC-inspired "complete environment" application towards the "bundle of tools" application.

- Object-oriented programming environments leveraging bitmapped display technology to simplify program development and integration. One of the earliest and most forward thinking developments by early bitmap display researchers was in the area of object-oriented programming languages, such as *Smalltalk*, which exploited a rich display environment. Without more elaborate display systems, many object-oriented programming environments become too cumbersome for applications beyond that of the early artificial intelligence applications intended. The evolution of object-oriented programming environments for widespread use, coupled with the availability of ubiquitous bitmap display technology will be the basis for the development of new and more complex applications technology.

Some new applications of this technology should also be noted:

- A single multi-window display can replace many terminals. A terminal emulator or editor per window can be used to provide access to a dozen or so "terminal screens."
- Items, such as interactive instruments (e.g., swing needle gauges and meters, chart recorders and bar graph displays), are used to monitor and control computer systems. Interactive instruments are merely programs which are intimate with the window system and mouse instead of with character keyboards and displays.
- Windows are used to separate environments on different hosts and in different programming environments. A popular way of gaining access to PC-applications on a workstation is through "MSDOS in a window." One can then, for example, cut and paste a LOTUS Spreadsheet onto a UNIX-based CAD package. On a single display, VMS, UNIX, and MSDOS may all be represented, somewhat like an ecumenical meeting of religions.
- And finally, the redefinable nature of some of the new window system environments can be employed to design a person's "ultimate environment." In effect, one can reprogram the "human interface" of the window system to suit.

Along with increased flexibility and power, the redefinability of bitmapped window systems provide a much greater challenge to the programmer. Object-orienting programming environments may become the only viable way of meeting this challenge in the long term.

It is important to note that, while Xerox PARC started the ball rolling with its excellent desktop metaphor, other metaphors are possible. The current industry investment into modern display systems capable of **X** does not solely imply the trend of the moment, but is also the basis of an infrastructure which is malleable to future trends.

What is X Windows?

The *X Window System*™ (hereafter referred to as **X**) is one of the most widely used *Graphical User Interfaces* (GUI), or bitmapped-window display systems. It is supported by all major workstation vendors, and is in use by a few hundred thousand users worldwide.

While **X** predominates on UNIX systems, it is not specific to just UNIX systems. *DECwindows*, a version of an **X** windows system, is available on DEC's proprietary VMS operating system. Apollo and NeXT workstations also provide access to **X** through their own proprietary window systems. In fact, there is no reason why **X** could not be run in place of Presentation Manager under OS/2.

X offers more than just a raw environment—it also offers a platform for uniquely incorporated commercial applications packages. Any major vendor offering an application package now available on a bitmapped UNIX workstation should certainly be aware of **X** and be incorporating it into future product planning.

Motif and Open Look

In addition to writing applications software, some industry groups have created proprietary software packages and standards for interfaces which leverage the display capabilities of **X**. These packages are then integrated into applications to improve the "look and feel." The two most significant commercial packages in this area are the Open Software Foundation's *Motif* and UNIX International's *Open Look*.

continued on next page

X Windows *(continued)*

Both of these packages are currently competing for control of workstation "look and feel" standards.

A recent addition to these two packages, IXI's *X.desktop,* places a Macintosh "look and feel" display on top of **X**. This product appeals to those comfortable with the Macintosh interface and is a reassuring platform for software developers with prior Macintosh experience. However, as with all compatibility approaches, there are limits to this simulation—as there really is not a true Macintosh underneath the display.

Other competitors to these three packages are anticipated from other vendors. As a consequence, the customer will have to determine which standards and implementation are "the best."

An industry standard

While there is considerable debate among the workstation audience on whether **X** should be considered the "best" standard for the job, there is no question that **X** currently holds the lead as the standard window system. Its strength as the industry "adopted" standard will thus influence all future bitmap graphics products in the next decade. Indeed, **X** is to window systems what FORTRAN was to computer languages in the 60's–70's.

X continues to facilitate the widespread acceptance of bitmapped graphics UNIX workstation environments. The rapid rise of the workstation was a key turning point in the evolution of the computer industry, with repercussions in future design goals and product offerings. One can never emphasize enough the importance of leveraging flexible standards to enhance a product's acceptance throughout the industry.

Background: Objectives, Tradeoffs and Architecture

Prior to **X**, most early window systems were embedded in the operating system supervisor layer, resulting in unanticipated problems. Supervisor-based systems, while faster in immediate response time, are difficult to debug, costly to implement, support, and improve, and are frequently ignorant of the advantages of networks.

In addition, most of these systems used dedicated window managers and applications programmer interfaces, each with different structures and motivations. This resulted, of course, in varying degrees of success and failure in handling the issues of information representation. Indeed, the differences between window systems was so large that in some cases it made software ports between window systems more onerous than recreating the entire program under the target window system from scratch.

Though limited, these early bitmapped display systems, developed at universities and research labs like Xerox PARC, fostered the development of new window systems technology.

History of X

X was the brainchild of Robert Scheifler, Jim Gettys, and others at MIT, as part of *Project Athena*, a research project devoted to the examination of very large networks of personal computers and workstations. As part of this study, a unifying window system environment extending over all systems was deemed necessary. **X** was envisioned as this window system—one that could be used among the varied heterogeneous computer architectures and networks.

As Project Athena progressed, **X** was evolved into a portable network-based window system. Much of the early work on **X** was derived from an extant Stanford window system called **W**. In fact, the name **X** was simply a play on the previous work **W**. To this day, the name **X** has stuck in the popular parlance.

Current **X** releases contain two numbers: the *version number* indicating major protocol or standards revisions, and a *release number* indicating minor changes. As of this date, the latest version is X11 Release 4, also known as X11R4. Major revisions of **X** are incompatible, but there is backwards compatibility with minor releases within major revision categories.

Present day window systems

With the introduction of Apple Macintosh and Microsoft Windows products, decisions and tradeoffs were biased towards certain marketing goals. The Macintosh focussed on ease of use and immediate functionality. This was achieved by avoiding the keyboard and using the mouse as much as possible. As a consequence, all programs on the Macintosh must use this same interface style, at the cost of more complex functions.

Microsoft Windows on a PC (identical to *Presentation Manager* under OS/2) was molded by a more pragmatic goal—the need to consistently and accurately migrate a huge MSDOS applications base onto a bitmapped environment. This was done at the cost of appearance and some functionality.

Look and feel

Both of these window systems stress sameness of "look and feel" as a virtue in all contexts—neither is in search of new functions (e.g., extensibility), and configurability is not a major concern. Should either firm have ultimate control of the most straightforward function and appearance recipe, undue influence could be exerted over all kinds of products developed by other vendors. This attempt at control, dubbed the "look and feel wars," has resulted in much litigation and discussion of copyright and patent rights in the software display arena. This attempt at control is tantamount to holding patent rights on the lever—and obviously upsets quite a few people.

NeWS

Other window systems of interest, like Sun's *NeWS* (Network Window System) and NeXT's Display PostScript-based system, offer a grander scale of window system functionality. NeWS, much like its ancestor *Andrew*, is a highly extensible network-based window system. With NeWS, an object-oriented *PostScript*, interlinked with **C**, allows for elegant design that leverages the best properties of both languages. But this is done at the cost of requiring programming expertise in two languages. One might say that in comparison to **X**, NeWS uses PostScript as a replacement for Xlib and **X**-Protocol, and is conceptually simpler.

NeWS has many ardent supporters in the PostScript crowd, but is of less interest to those who prefer to let applications programs write to the laser printer. NeWS has also been accused of being both slow and huge by **X** advocates.

However, NeWS is available on other workstations including Suns. Like **X**, NeWS can also operate as a replacement to Presentation Manager on OS/2. This potential independence of operating system environment is termed an "open standard" by Sun, and should not be confused with "publicly available" **X**.

continued on next page

X Windows *(continued)*

XView

For those who use *SunView*, a more archaic proprietary window system, Sun has introduced a product (*XView*) which purports to tie them all together. The result—in the grand tradition of Sun software, a gargantuan agglomeration of a window system that effectively sings and dances all the tunes at once, as well as providing a reason for selling more disk drives and RAM memory. To be fair, once one enters the bitmap display world, one must commit to substantial storage volume anyway. I suppose one might refer to this as "storage inflation."

NextStep

With NeXT's window system, *NextStep*, the heritage of the Macintosh is immediately evident, with the window system itself based on *Display PostScript*. But there is a twist. The strongest advantage to NextStep is not even in the window system proper. It is, instead, *Interface Builder*, an application that permits interactive design of a program's graphical appearance and controls. After an acceptable interface has been created, Interface Builder then generates the appropriate code to be embedded into the new application. Used properly, this feature can save considerable time and expense in software development.

Recently, in an attempt to be "supplier-independent," NeXT persuaded IBM to offer its NextStep bitmapped environment on IBM's new exotic RISC workstations, alongside the version of **X** already available. As NextStep supports a very complex development environment (some believe too complex for most customers), the use of Interface Builder by software developers will be key to eventual customer acceptance.

Both NeWS and NextStep attempt to entice the customer with increased versatility. The question each company must ask is "Can the customer handle the greater variety without becoming confused and frustrated?" The answer will be determined in the marketplace.

But what about X?

All of the previously discussed window systems have been carefully engineered to fit goals established by each individual vendor. Well, what about **X**? Since **X** is not owned by any single vendor, there were no overriding marketing goals or corporate motivations which shaped its design. Instead, **X** is a bit like Topsy—it was just "growed."

What it is *not*

X is best described by what it is *not*: **X** is not based on any set graphics architecture; it is not licensed by any firm; it does not have a mandatory applications interface, appearance, or window manager; it does not rely on the prescience of a given operating system. In short, **X** is a window system that presumes very little knowledge and permits choices to be made outside of its context. This is **X**'s great strength, in that it can be made to do almost anything, as well as its great weakness, in that everything else must be specified—no small task.

This Zen-like perspective on the world has spawned much controversy, with detractors marking it as a cumbersome, inelegant albatross, and advocates heralding it as a universal standard that will unify the diverse graphics display market for applications use. To some degree I believe both are right. Thus my analogy "**X** is the FORTRAN of window systems."

Sample session with X

For those unfamiliar with **X**, a sample session is in order. This example assumes the basic X11 release. Other **X** window versions, such as *Motif* and *Open Look*, will differ somewhat in appearance and function.

Upon login to a display terminal, a UNIX shell runs in a terminal emulator window (an *xterm* client). This window can then be used as if it was an ordinary CRT display terminal. However, **X** offers more interesting variations, as we shall see.

A window manager is now invoked. While windows can be created without a window manager in place, a window manager permits windows to be resized, moved, and otherwise modified on demand. Various window managers are available, and each can be extensively tailored much in the manner that UNIX shells can be customized with start-up scripts.

Once the window manager is operating, a press of a mouse button summons a pop-up menu, while pulling the handle selects the desired functions, such as moving and resizing the login window to the side of the display. Other mouse selections permit the creation of a clock display or various terminal emulator windows (some of which run on other hosts in our network—see Figure 1).

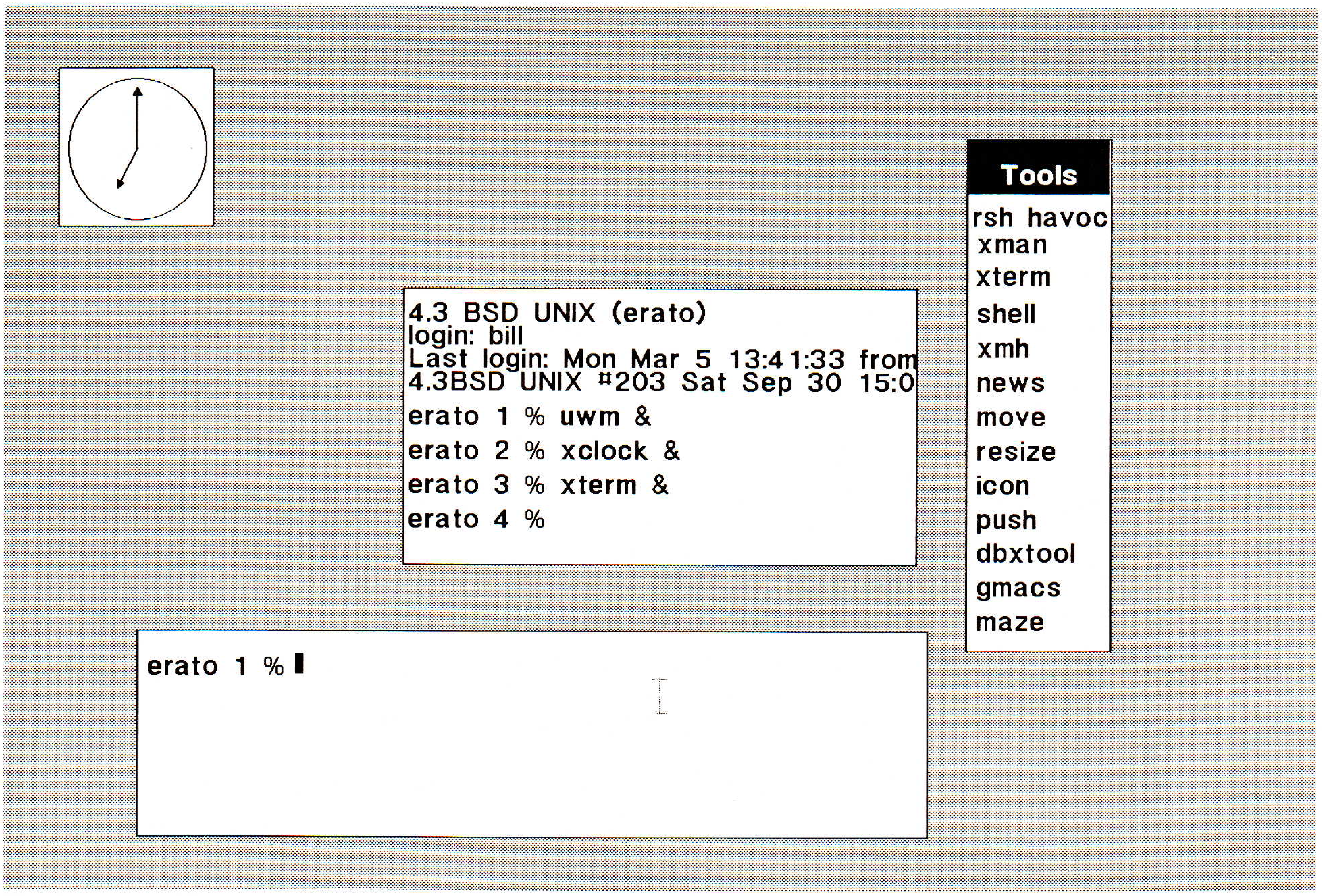

Figure 1: Sample **X** screen

By pointing to each of these windows with the mouse, one can indicate on which window our keyboard is "focused." By clicking on a terminal window, options may be altered and a scroll bar on the side of a window can be selected.

continued on next page

X Windows *(continued)*

Due to the interactive nature of **X** windows, a simple "talky" description fails to capture its rich nature and abilities. For example, **X** windows can allow a graphics application to create many different windows and "widgets" on demand. With a color display, background and text colors can be altered to suit any need.

Breakdown of X Windows

Conceptually, **X** consists of the following parts: an **X**-server, **X**-client, **X**-protocol, Xlib, toolkits, and window managers (see Figure 2).

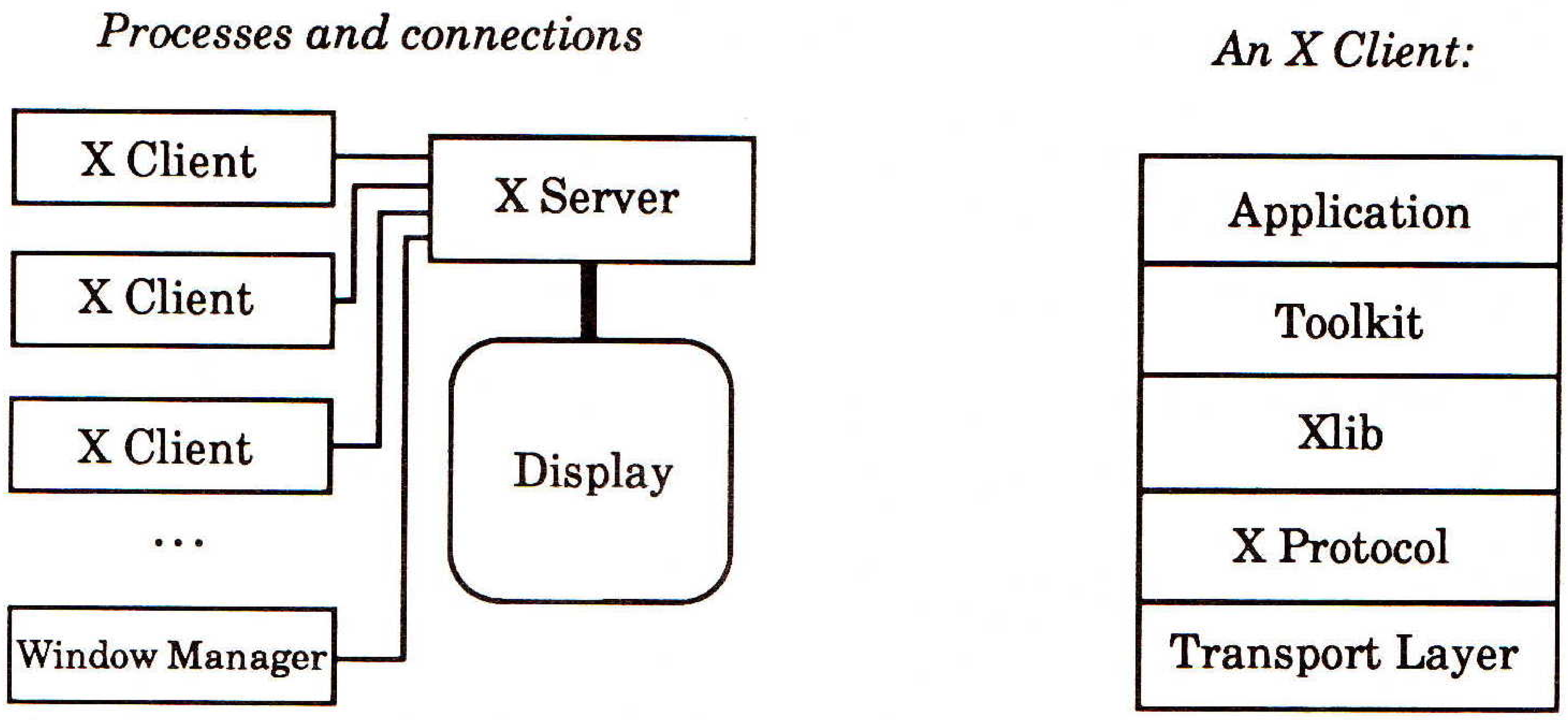

Figure 2: **X** System model

Client–Server model

The **X**-*server* is a dedicated program which provides display services on a graphics terminal, on behalf of a user, at the request of the user's **X**-client programs. An **X**-*client* is an application program that is intimate with **X**. Typically, many **X**-clients compete for the services of one **X**-server per display per user. Conflicts for service are resolved by the window manager, a separate entity altogether.

Server and client(s) communicate over a network or IPC connection. Thus, **X** is a network-based window system (like NeWS). The **X**-protocol, running within the network connection, allows request-response between client and server. It is a very simple protocol, relying only on an outer transport layer for a point-to-point reliable connection between a given client and server. The **X**-protocol is usually run over either a TCP stream, a UNIX IPC connection, or a DECnet connection (on VMS). However, there is nothing to prevent it from being used over other protocols, such as OSI TP4 or ISDN LAPD.

Xlib

The rudimentary programming interface for the application (the "nuts and bolts") is contained in *Xlib*. Xlib is a collection of primitive subroutines embedded in all **X**-clients, and can be considered a kind of "ground layer." It is possible to write applications programs entirely with Xlib. In fact, most existing **X**-clients are or were developed in this fashion. However, this approach is not trivial.

Toolkits

With *toolkits*, the **X** window systems model becomes much more abstract and object-oriented. Unfortunately, or fortunately as the case may be, there is not just one but many toolkits available which afford alternative applications interfaces, screen appearances and functions. Many manufacturers and universities each have there own toolkits.

The "standard" **X** toolkit available was called *Xt* in X11R3 and prior releases. With the recent release of X11R4, Xt has evolved into *Xaw*, more commonly referred to as "Athena Widgets." In a sense, Xaw is the defacto "standard" toolkit.

Programming a window system for a new application becomes much less herculean with the use of toolkits, but it may still baffle even accomplished programmers. In a recent conversation with a Berkeley UNIX programmer and long-time user of **X**, he attempted to add a trivial feature to the client *xbiff*, only to find after a protracted period that a substantial amount of "source education" was required. End of addition of new trivial feature! The value of NeXT's Interface Builder now becomes quite apparent, as most of us are more interested in an end product rather than in wrestling with the arcane details.

X is unique in offering a choice of arbitrary window managers and allowing for different strategies and policies within the window system. Again, this is another example of the **X** philosophy of: "If you don't like it, redesign it to suit your own needs or taste." Indeed, by simply altering the toolkit and window manager, much of the appearance and function of **X** can be changed. And, as a window manager is run like any other client program, it can be, if needed, run on a distant host over the network. In fact, one can even run **X** without a window manager if one so desires, or even switch to a different window manager on the fly.

Software products

There have been many versions of **X**. With the current version of **X**, version 11, the commercial market has begun to grow in significance. And, as one might expect, with the rise of interest in **X** there has been an increase in **X** related products, in both software and hardware.

Software products for **X** run the gamut from basic client/server packages to application clients and utility clients, with new products appearing daily. It is important to note that, since **X** is a network-based window system, it is possible to run application clients off of other hosts, keeping the display host free for other tasks. Thus, a single copy of software purchased for a particular network host can be used on all other hosts running **X**-servers (displays).

To use **X** minimally at a given facility, a set of **X**-clients and Xlib for participating mainframes without displays, or an **X**-server per computer with a display, is required.

Application products

Many applications software vendors have also announced support for **X** in their existing products. However, the majority of these products have not yet been released or are available only in experimental form. (A notable exception is Frame Technology's *FrameMaker* electronic publishing package.) These, and other yet to be announced product offerings, are anticipated for a 1990 release.

Some expected products include spreadsheets like Access Technology's *20/20*, and INFORMIX's *WingZ*. There is the usual rumor of Lotus Development Corporation's interest, but let's not hold our collective breath. Other products, like RTI's *Ingres* database, and BBN's *Slate* office automation package are also promised in the **X** market.

An unusual product similar in intent to NextStep's Interface Builder is Integrated Computer Solutions *Builder Xcessory*. This interface builder for **X** permits graphical design of the interface, while it spits out the code to make the interface work with a program.

continued on next page

X Windows *(continued)*

If this becomes a viable product, and if it lives up to its expectations, Builder Xcessory may become the hot item for **X** product development.

X Terminals: Fact or Fancy?

X is typically used on an existing workstation, with the **X**-server running as an ordinary process. However, for those who do not wish to own workstations or who must share existing processor resources, **X** terminals are available. These terminals are actually dedicated **X**-servers that connect (via serial line or Ethernet) to hosts running various **X**-clients (applications). **X** terminals are available from Network Computing Devices, Tektronix and Graphon, to name just a few. IBM, DEC, HP and many large Japanese firms have just introduced a plethora of **X** terminal products in early 1990.

One may wonder if it is wise technically to use bitmap display systems like **X** terminals, or does the inherent data bandwidth limit get in the way of daily use? For low-end monochrome or 2-D graphics, **X** terminal bandwidth limitations are not particularly important. However, for the high-end color and 3-D systems, these bandwidth limitations begin to assert themselves. As the amount of information contained in the image skyrockets, the communications technology gets in the way, resulting in significant degradation of function and speed. We have yet to see **X** terminals with sufficiently high bandwidth, such as FDDI would provide.

X terminals are usually priced below that of workstations, in the $2,000 to $7,000 range, and require large amounts of RAM (2–8 megabytes) if extensive use of the windowing system is made. Large amounts of memory are critical to providing a "backing store" as windows are overlapped or covered, since (unlike running **X** off of a virtual memory-based workstation) the **X** terminal must contain as much RAM as needed during peak use. As a consequence, many users of **X** terminals find the standard (512KB) memory configuration available from the vendor to be insufficient.

Memory needs

The large memory requirements inherent in heavily-used **X** terminals is of major concern to both the customer and the vendor. Graphon is a vendor that has approached this problem in a novel way. Graphon's **X** terminal ($1000 to $2000), offers a proprietary **X**-server run on the UNIX host, and a proprietary serial protocol which communicates with the **X** terminal. Dynamic memory needs are handled by the UNIX virtual memory system to the degree required to backup the display. The drawback to this arrangement is that the **X** terminal is slower. Additionally, one becomes dependent on proprietary licensed Graphon software, assuming it is available for a particular system. (It's not available on many, including the one I am working on today).

An unusual **X** hardware product from HP is a TGA (Texas Instruments Graphics Architecture) display board that uses a TI Graphics processor to run an **X**-server on the Graphics processor. A high-performance **X** environment can then be run off of a PC without overloading its limited capabilities. (Fortunately, many current PCs are not as limited as those of a few years ago). A footnote here; the TI Graphics Processor 34010/34020 is frequently showing up in high end **X** products (i.e., the IBM **X** terminal), as is dual-port VRAM (video RAM—special memory devices that reduces the cost of display refresh while preserving high speed manipulation of the screen by the server). A processor specially designed for bitmaps, and low arbitration cost display memory are the hallmarks of high-end display systems.

Software packages

There are also many software packages available which turn IBM PCs and Apple Macintoshes into **X** terminals. Graphics Software Systems of Oregon and Integrated Inference Machines of Southern California offer PC to **X**-server software, while White Pine Software of New Hampshire provides an **X** product for the Macintosh. As other firms follow with **X**-server software packages, this product will become as common as terminal emulators for PCs and Macintoshes are now. In fact, it may be the case that terminal emulator software packages for the PC may eventually include an **X** terminal emulation as part of the package.

Are X Terminals sensible?

Some find the concept of **X** terminals retrograde, because, after all, the trend towards decentralized computing implies putting the processor onto the desktop. However, given the typical cost of a bottom end ANSI terminal (approximately $500) and a good bitmap workstation (approximately $10,000), a twenty-fold difference in price tends to support the notion of a market niche in **X** terminals, especially when current software and environments don't utilize workstation speeds to their ultimate capabilities. In fact, the eventual commoditization of **X** terminals (like ANSI/ASCII terminals) may result in a similar fate for low-end workstations.

An opposing view cites **X** terminals as more "correct" in fitting a network model for decentralized computing, with **X** terminals considered "display clients" for the use of "compute servers." Thus, **X** becomes a graphics network service on the presentation and application levels.

X terminals are considered controversial for philosophical as well as technical reasons, and even market researchers like Dataquest have yet to decide whether they constitute a new niche market, or are just a flash in the pan. Since market estimates vary so widely, most customers have held off on purchase, and are instead waiting for purchase and manufacturing commitments from large companies and vendors to validate these products.

Are PC X Terminal Emulators sensible?

A humorous analog to the "workstation verses **X** terminal" debate is the "**X** terminal emulators for PCs and Macintoshes versus the **X** terminal" debate. Some state that the inherent limitations of PCs make them insufficient for use in displaying **X**, and that the specialized hardware of **X** terminals is required for adequate **X** performance. There is a factual basis to this argument, since the basic standard PC was never intended for elaborate graphics, and no amount of software can make up for hardware that is not present.

However, PCs have evolved considerably since then, with much of this development in the display arena. Thus, many high-performance PCs can provide adequate **X** performance. In fact, much of the special display hardware used today in bitmap graphics was developed for and sold in the high-end PC display systems market! One could say that the evolution of the PC and workstation permitted the development of the **X** terminal market.

As **X** breathes life into the high-end terminal market, one can already observe the price erosion inherent in the computer industry. Eventually, **X** terminals may displace the ANSI terminal itself. **X** terminals may even be offered with the ability to field upgrade to a full workstation!

continued on next page

X Windows *(continued)*

The future of X

Among users of window systems, the growth in the use of **X** has been spectacular. **X** has evolved from one of many competing standards in the UNIX workstation arena to that of the major player in windowing system. It has become almost mandatory that workstations support **X** as well as any other proprietary window systems. Again, this is similar to FORTRAN requirements on scientific computers in the 60's and 70's.

In recognition of this success, Project Athena has turned over control of **X** to the *X-Consortium*, a group of companies who are attempting to refine standards for the commercialization of **X**. It is hoped that under "benign" oversight, **X** will no longer carry the stigma of an unruly university curiosity. Whether this consortium, or some other group, will accomplish this goal remains to be seen.

As **X** has reached critical mass as a standard for applications, excitement in the market has increased. Applications vendors have already announced commercial products using **X**. The first wave of commercial **X** products is expected within the year.

At the same time, vendors from rival groups (Motif from OSF and Open Look from UI) have introduced competing "superstandards" for "look and feel" in the hopes of eventually controlling the commercial **X** market. The jury is still out on both of these systems. Ultimately, the customers will vote with their pocket books, and **X** will evolve again.

Additional reading

In the preparation of this article, the following references were found useful:

"X11R3 Release Documentation" by **X** consortium staff and contributors. (Unpublished).

"O'Reilly & Associates **X** series" Volumes 1–3 by Nye, Querica, & O'Reilly, O'Reilly & Associates, ISBN 0-937175-26-9, 1988.

"The NeWS Book" by Gosling, Rosenthal & Arden, Springer Verlag Press, ISBN 0-387-96915-2, 1989.

[Ed.: See also **X** book reviews in *ConneXions*, Volume 3, No. 10, October 1989.]

BILL JOLITZ was the CEO of Symmetric Computer Systems for over five years, and has been responsible for strategic marketing for ISDN products, as well as specialized TCP/IP networking products. He has a broad background in business management, product management, and product design. he also has substantial experience in both the UNIX kernel and TCP/IP networking, and was principal developer of 2.8 and 2.9 BSD at the University of California at Berkeley. He was the chief architect of National Semiconductor's GENIX project which was the first virtual memory microprocessor-based UNIX system. He is currently affiliated with TeleMuse, a market research firm specializing in the telecommunications and electronics industry.

Upcoming Events

IEEE INFOCOM '91—The Conference on Computer Communications Networking in the 90's will be held April 7–11, 1991 at the Sheraton Bal Harbor, Miami Florida.

Call for papers

Authors are invited to submit full papers on recent advances in computer communications. Areas of interest include, but are not limited to:

Integrated Services Networks	Network applications
Metropolitan & Wide Area Networks	Host interface Architecture
Local Area Networks	Photonics in Communications
Network Management	ISDN Technology
Communication Protocols	Broadband ISDN
Distributed Network Algorithms	Network Interconnection
Network Modeling & Analysis	Network Reliability
Satellite Networks	Network Routing & Addressing
Wireless Data Networks	Computer Security & Privacy
Networks Design & Planning	Multimedia Information
Systems Switch Processor Architecture	Network Standards

Important dates

Full paper (5 copies):	August 1,	1990
Notification of Acceptance:	October 31,	1990
Camera-Ready Copy:	December 15,	1990
Conference:	April 7–11,	1991
Tutorials:	April 5–6,	1991

Submit papers to:

Dr. N. Shacham
Technical Program Chairman, *IEEE INFOCOM '91*
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
Telephone: 415-859-5710 Email: `Shacham@sri.com`

A Letter to the Editor

Hi Ole,

Knowing your penchant for collecting such things, I thought I'd let you know that I've just finished rewriting my "Network Manager's Reading List." This is version 2.0 of a list I originally created in 1988.

It contains around 30 different books and resources of use to network managers using TCP/IP, UNIX, and Ethernet. Each item is annotated, has complete access information, and introductory information excerpted.

You can find a copy via anonymous FTP from `emx.utexas.edu`. It's located in the `pub/netinfo/docs` directory as:

```
network-reading-list.ps
```

for the *PostScript* version. There is also an ASCII text version.

Cheers,

Charles E. Spurgeon, University of Texas, Austin.

Thank you for the information! —Ed.

Fraternizing with the Enemy
or
ANSI and IETF Find Common Ground

by Lyman Chapin, Data General

First meeting

On April 23 and 24, 1987, the Internet Engineering Task Force (IETF) and ANSI-accredited task group X3S3.3 (responsible for the network and transport protocols of the Open Systems Interconnection (OSI) architecture) met jointly for the first time at Bolt, Beranek & Newman in Cambridge, MA. In 1987, the two groups knew very little about each other; only one or two people were active in both, and information and ideas were only sporadically shared between them.

In December of this year the IETF and X3S3.3 will meet jointly for the second time, under very different circumstances. In 1990, a large number of people are active in both groups; wholesale information exchange between X3S3.3 and the relevant IETF working groups is taken for granted; and the two groups are engaged in cooperative efforts on routing, protocol architecture, and naming. Until recently, an IETF routing expert served as vice-chairman of X3S3.3; last November, the X3S3.3 chairman was appointed to the Internet Activities Board (IAB).

Common ground

There is still a part of the IETF community that reflexively scorns anything tainted by association with "OSI," but the imminent advent of a multi-protocol internet of OSI and TCP/IP has focused most people's attention on the substantial (and growing) common ground that is shared by ANSI and the IETF.

The ANSI-accredited standards committees concerned with networking (principally X3S3, *Data Communications*, and X3T5, *Open Systems Interconnection*; hereafter, simply "ANSI") share many basic goals and interests with the working groups of the IETF; both, for example, try to promote interoperability in large, heterogeneous internets by establishing agreements among technical experts on standard protocols, addressing schemes, and management systems.

Charter and scope

The charter and scope of the two groups, however, differ significantly. ANSI is entirely a standards-making body; it is not directly involved in the application of the standards it produces. The IETF is also concerned with standards, and is itself a standards-making body, but its responsibility for "the Internet"—the globally interconnected collection of networks and hosts that communicate using the TCP/IP protocol suite—is much broader (and much less formal) than ANSI's responsibility for OSI.

ANSI's job is to specify standards; the IETF's job is to guide the operation and evolution of the Internet. The IETF's scope, however, is narrower than ANSI's. The IETF is concerned with (and can therefore focus exclusively on) one particular (albeit pervasive) internet. ANSI, on the other hand, is responsible for the development of hundreds of international standards for OSI which must reconcile (or at least accommodate) the technical and political requirements of 28 countries, in which every conceivable approach to networking is expressed. These differences of charter and scope necessarily affect the way in which the two groups operate.

They have also, in the past, had the unfortunate side effect of inhibiting cooperation, by obscuring the common ground that exists with respect to many technical—and even some political—networking issues.

Routing

The extent of ANSI and IETF collaboration today is impressive by the standards of just a few years ago. Both ANSI and the IETF are working on intra-domain and inter-domain routing. (Perhaps inevitably, the two groups do not share a common nomenclature. The OSI "intra-domain" routing corresponds roughly to internet IGP-style routing within an autonomous system; "inter-domain" routing to inter-autonomous system EGP-style routing; and "intermediate system" to "gateway," or more recently "router").

ANSI's intra-domain routing protocol, referred to as "IS-IS" (for "intermediate-system to intermediate-system"), has been taken up by one of the IETF working groups; an inter-domain routing protocol developed by another IETF working group (BGP, for "Border Gateway Protocol") has been taken up (in the form of a variant called the "Border Router Protocol") by ANSI task group X3S3.3.

IP over *x*

With substantial joint participation, both ANSI and the IETF are working on specifications that define the operation of an internetwork protocol (the RFC 791 IP, in the case of IETF, and the ISO 8473 IP, in the case of ANSI) over specific individual networks, such as FDDI fiber-optic LANs.

Network Management

ANSI and the IETF are both working on managed object definitions and the specification of management protocols, sharing expertise and specific proposals. And both groups have demonstrated a strong desire to exploit the many other opportunities that exist for joint development efforts.

What next?

It is still too early to predict that a harmoniously integrated community of OSI and TCP/IP networking will be the outcome of these initiatives. The success of the ANSI/IETF collaborative efforts is seriously threatened by the persistence of a strong individual and collective anti-OSI bias within the IETF, and by the slow pace of ANSI standardization, which discourages IETF members (accustomed to much quicker evaluation and resolution of issues) from becoming involved in ANSI activities.

The imperatives of a multi-protocol Internet, however, argue convincingly for cooperation—as do most of the ANSI and IETF participants. OSI will be an important part of the Internet; Internet engineering expertise will be an important factor in the successful deployment of OSI. ANSI has one, IETF has the other. The connection, as they say, is obvious.

A. LYMAN CHAPIN received his B.A. in Mathematics from Cornell University in 1973. Since 1977 he has worked at Data General Corporation as a designer and implementor of communications and networking systems, and is currently a Senior Consulting Engineer in the Systems Architecture Group. He is the chairman of the American National Standards Institute (ANSI) task group responsible for the development of OSI Network layer and Transport layer service and protocol standards (X3S3.3), vice-chairman of the Association for Computing Machinery special interest group on data communications (ACM SIGCOMM), and a member of the Internet Activities Board (IAB).

The Telnet LINEMODE Option

by David A. Borman, Cray Research, Inc.

Introduction

The Telnet LINEMODE option provides a standard, well defined method for moving the bulk of character input processing to the local Telnet application, thus reducing the number and frequency of packets sent over the network, and the user's frustration at having to wait for single characters to be echoed remotely over a slow and/or congested network.

Why LINEMODE?

The LINEMODE option provides for faster and more efficient use of the network, but in order to understand why there is a LINEMODE option, it is necessary to understand a few things about the Telnet protocol and how it has traditionally been implemented. The primary goal of the Telnet protocol "...is to allow a standard method of interfacing terminal devices and terminal-oriented processes to each other." [1, 8]

The Telnet RFC also states: "...the keyboard produces outgoing data which is sent over the Telnet connection and, if 'echoes' are desired, to the NVT's printer as well. 'Echoes' will not be expected to traverse the network (although options exist to enable a 'remote' echoing mode of operation, no host is required to implement this option). [2]

Insofar as the availability of local buffer space permits, data should be accumulated in the host where it is generated until a complete line of data is ready for transmission..."

The motivation for this rule is the high cost, to some hosts, of processing network input interrupts, coupled with the default *Network Virtual Terminal* (NVT) specification that "echoes" do not traverse the network. [3]

Suppress Go Ahead

This implies that Telnet is more of a line oriented protocol. However, in today's world of high speed local area networks, with fast processors where processing network interrupts is inexpensive, it is reasonable to have the remote machine controlling all the character input and echoing. The local Telnet program would like to just send all the user's data verbatim to the remote machine, and not worry about any of the character processing. This can be achieved, using the Telnet *Suppress Go Ahead Option* (SGA). The RFC which describes this option states:

"As the SUPPRESS-GO-AHEAD option is sort of the opposite of a line at a time mode, the sender of data which is suppressing GO AHEADs should attempt to actually transmit characters as soon as possible (i.e., with minimal buffering) consistent with any other agreements which are in effect.

In many Telnet implementations it will be desirable to couple the SUPPRESS-GO-AHEAD option to the echo option so that when the echo option is in effect, the SUPPRESS-GO-AHEAD option is in effect simultaneously: both of these options will normally have to be in effect simultaneously to effect what is commonly understood to be character at a time echoing by the remote computer." [4]

Thus, many implementations of Telnet in the past have just negotiated for SGA and ECHO [7], and gone merrily on their way doing single character, remote echo I/O, totally ignoring the default line-at-a-time mode of the NVT as described in the Telnet protocol.

The changing face of networking

However, over a low speed or highly congested network, all this single character I/O can add to congestion, and frustrate the users as they wait for the remote machine to echo the typed data. There are also machines (like supercomputers), where the cost of handling network interrupts still has a significant effect on the performance of the machine, and which would prefer to not be bothered with doing single character terminal I/O. These machines would rather just let the front end machine deal with that. Also, there is more and more talk in some areas about charging for network use on a packet by packet basis. So, in today's world, we are once again seeing a need to return to line at a time mode, though for different reasons than originally envisioned in the Telnet RFC.

The problem with line-at-a-time mode as described in the Telnet RFC is that the Telnet RFC does not describe line-at-a-time mode in much detail. It says nothing about where the input line editing functions should take place, whether they should be processed on the front end, or just embedded in the line that is sent to the remote machine. To answer these questions, and to provide missing functionality, the LINEMODE option was born.

The LINEMODE option provides a standard method for doing line at a time mode Telnet, and doing all the line editing functions in the local Telnet application (i.e., the client), so that only edited lines are sent to the remote system. This provides many advantages:

- The number of packets sent over the network is decreased.
- The user only sees network delays on a line by line basis, not character by character.
- The user may decide whether he wants the local or remote special character mappings. This is very useful when connecting between heterogeneous systems.
- The local Telnet application knows which typed characters should be mapped into which Telnet commands.

History

The first LINEMODE implementation was done at Cray Research, Inc., in the spring of 1986. At that time, the goal was to implement it within the existing framework of the Telnet protocol. Since the RFCs gave passing reference to "line-at-a-time mode" without defining its characteristics, and the SGA RFC talked about using SGA and ECHO to do "character-at-a-time remote echoing," it was decided that no SGA and no ECHO would mean LINEMODE. Further, since the ECHO option would need to be enabled and disabled (for typing passwords), LINEMODE became dependent on just the state of the SGA option.

This initial LINEMODE implementation was shipped with UNICOS 1.0 from Cray Research, Inc., and at the last minute the client side was included on the 4.3BSD release from Berkeley. This initial implementation worked well, but implementing it on the SGA option was clearly not the right way to do it. Besides, there were deficiencies in the implementation. If the user changed any of his special characters on the remote side, there was no way to propagate this information to the front end. The list of Telnet commands was also missing some needed values; *suspend*, *abort* and *EOF* were the most obvious ones that were missing.

continued on next page

The Telnet LINEMODE Option *(continued)*

In the spring of 1988, a proposal for a LINEMODE option was brought to the IETF (Internet Engineering Task Force). There was sufficient interest, and an IETF working group was formed. By the summer of 1989, and many revisions later [5], the LINEMODE option was fairly well defined. [6] The described protocol is quite different from the original proposal, but the functionality has been expanded. After an initial implementation discovered some flaws in the protocol, the draft RFC was modified one more time, and then submitted for RFC status. In August of 1989 it was released as RFC 1116. The old LINEMODE based on the SGA option is now usually referred to as "kludge LINEMODE" and the LINEMODE option is referred to as "real LINEMODE" or just "LINEMODE."

Features

The LINEMODE option has many features:

- Character processing is moved from the Telnet server to the Telnet client.
- Echoing, line editing, and special character mapping is dynamically modified by the server. Thus, when an application is started that doesn't want remote line editing, the Telnet server can automatically detect this change in terminal state, and propagate the information to the front end.
- The translation of special characters (like interrupt process) to their Telnet commands is separate from input line processing. This allows the server to individually control these options. (This wasn't possible in the kludge LINEMODE implementation.)
- The special character mappings may be sent in both directions. This allows the user to choose either the local or remote character settings, and as characters are changed on the remote side, immediately reflect that change on the local side.
- Three new Telnet commands are added: ABORT, SUSP, and EOF. (The kludge LINEMODE implementation used BRK for ABORT, and some implementations used EOR for EOF.)
- The ECHO and TOGGLE-FLOW-CONTROL options are required for LINEMODE. This functionality is necessary, but there is no reason to have more than one method for doing these things.
- There is a provision for the server to send the client a bitmask of all characters that it is interested in. When any of these characters is typed, the client will immediately forward all currently buffered characters, and the special character. This allows the server to still handle any special characters that do not have Telnet command equivalents.

Non-features

There were many goals for the LINEMODE option, but there were also some specific areas that were explicitly chosen to be *not* addressed by the LINEMODE option. The main non-goal of the LINEMODE option is to allow the front end to exactly mimic what the back-end terminal driver would do. There are two reasons for this. The first is that it is a lot of work, and would require a lot more state information to be passed back and forth. It also involves a lot of system dependent parameters, and the NVT is supposed to be system independent. The second reason is that one of the goals was to define the LINEMODE option in such a way that the client Telnet application would not have to have a terminal driver built into it; it would be able to use the system terminal driver to do all the work. This would be an impossibility if the goal was to exactly duplicate the terminal driver of the remote side.

The two main topics that are avoided by this decision are how to erase characters from the screen, and how to echo non-printing characters. The effect of avoiding these issues in the LINEMODE option is that the user will get the characteristics of the front end machine.

Futures

The LINEMODE option definition is fairly concrete. Any future changes to the LINEMODE option would probably be limited to extensions to the current protocol. Currently, RFC 1116 is a "proposed elective standard." It will be re-issued as a "draft elective standard" as some point in the near future, and then after at least the minimum waiting period of six months, it will be re-issued as an elective standard.

Availability

UNICOS 5.1 from Cray Research, Inc., had support for the LINEMODE option. 4.4BSD from Berkeley will have LINEMODE support when it is released. There is a publicly available implementation (the code that will be released in 4.4BSD) that can be retrieved via anonymous FTP from `ucbarpa.berkeley.edu`. Look for a file named `telnet.YY.MM.DD.tar.Z`, where `YY.MM.DD` is the year, month and date that that particular version of Telnet was made available for anonymous FTP.

Summary

The LINEMODE option provides a standard, well defined method for moving the bulk of character input processing to the local Telnet application, thus reducing the number and frequency of packets sent over the network, and the user frustration level at having to wait for single characters to be echoed remotely over a slow and/or congested network.

References

[1] Postel, J. & Reynolds, J., "Telnet Protocol Specification," RFC 854, p. 1.

[2] Ibid. p. 4.

[3] Ibid. p. 5.

[4] Postel, J. & Reynolds, J., "Telnet Suppress Go Ahead Option," RFC 858, p. 2.

[5] Those who attended and participated in the Telnet LINEMODE working group meetings include: Coleman Blake, David Borman, Jeffrey Burgan, Allen Cole, Allan Fischer, Mike Karels, Stuart Levy, Louis A. Mamakos, Drew Perkins, Philip Prindeville, Joyce Reynolds, Bruce J. Schofield, David Wasley, and Bill Westfield.

[6] D. Borman, Editor, "Telnet Linemode Option," RFC 1116.

[7] Postel, J. & Reynolds, J., "Telnet ECHO Option," RFC 857.

[8] Shein, B., "The Telnet Protocol," *ConneXions*, Volume 3, No. 10, October 1989.

DAVID BORMAN has worked at Cray Research, Inc. since December of 1985, and is the project leader for the TCP/IP networking code that Cray Research releases with its UNICOS operating system. Prior to working at Cray Research, he worked for two years at Digital Equipment Corporation in Merrimack, NH, as a kernel developer for DEC's ULTRIX-11 product. He is currently an active member of the Internet Engineering Task Force, where he chaired the Telnet LINEMODE Working Group, and is currently the chair of the new Telnet Working Group. He received his B.A. in Mathematics, with a concentration in Computer Science, from St. Olaf College, Northfield, MN, in May of 1983.

Profile: CREN—The Corporation for Research and Educational Networking

by Mark Laubach, Hewlett-Packard Company

Introduction

This article presents a profile of what CREN is today with a glimpse of where it is headed. Numerous prior works have been incorporated into this profile, all liberally referenced. This article closes with a summary of technical challenges facing CREN for the future.

History

On October 1, 1989, the *Corporation for Research and Educational Networking* (CREN, pronounced "kren") was formed from the merger of the two already well known networks *CSNET* (The Computer+Science Network) and *BITNET* (The "Because It's Time" Network). The process was started in October 1988 when the boards of CSNET and BITNET voted to merge. The merger was later ratified by a vote of the BITNET member organizations in February 1989. The then newly elected Chairman and President of CREN issued the following statements to the new CREN membership [1]:

"The aims of CSNET and BITNET—to support and promote the use of computer networks on campuses and within research organizations—have converged over the last several years. We believe that by bringing these two networks into a single organization, we will be able to provide better service and more effectively participate in the fast-changing national network environment."
Bernard Galler
Chairman, CREN

"The growth of campus networks and the introduction of new technology make it necessary to build a common base of network services using the most progressive technology available. By eliminating the historical overlap between CSNET and BITNET, we will become more efficient, and more importantly, we can take a stronger role in the formation of the National Research Education and Network. We can achieve this goal faster and at a lower cost by leveraging the efforts of the two major academic networking organizations."
Ira Fuchs
President, CREN

Membership

CREN membership is currently open to accredited US universities, colleges, and two-year colleges; industrial US research labs having significant interaction with academia; and other institutions approved by the CREN Board [1]. Non-voting affiliation is currently available to other US or non-US academic institutions, industrial or government research labs, other government agencies, and other institutions approved by the CREN Board. Non-voting members are eligible for all BITNET and CSNET services.

Currently CREN's membership consists of approximately 130 CSNET service members and approximately 500 BITNET service members over 1750 nodes [3]. BITNET is further connected to several international networks supporting BITNET-style services:

Network	Location	Members	Nodes
EARN	Europe	650	800
NetNorth	Canada	110	180
ALNS, RUNCOL	Latin America	20	30
Gulfnet	Asia	65	100

This is a partial list, BITNET-style service networks are installed in approximately 35 countries worldwide.

Management

Over-all management for CREN is supplied by *EDUCOM*, a non-profit consortium for information technology in higher education, with headquarters in Washington, D.C. [1]. EDUCOM has filled this role for BITNET for the past three years. The BITNET Network Information Center is in the offices of EDUCOM. The *CSNET Coordination and Information Center* is at BBN Systems and Technologies Inc., in Cambridge, MA. The BITNET and CSNET networks will continue to have separate staffs and computer centers. Because the technologies of BITNET and CSNET are very different, CREN will concentrate first on combining administrative and user services functions.

Outreach

CREN is currently a member of the *Federation of Academic Research Networks* (FARNET) and is a participating member in the *Internet Engineering Task Force* (IETF) community.

Additionally, CREN has adopted an active policy for establishing connections to international cooperating networks and increasing its international membership. CREN maintains active cross-board participation with both *NETNORTH* (The Canadian Northern Network) and with *EARN* (European Academic Research Network).

Network connection services

CREN draws on both its historical BITNET and CSNET service families to provide a rich variety of network connection options:

•*RSCS/NJE over BISYNC*—The IBM *Remote Spool Control System* (RSCS) protocol is the means of connecting up BITNET services members. Traditionally, RSCS/NJE has been run over dedicated IBM BISYNC lines running at 9600 baud. This protocol supports rudimentary file transfer and remote command capability. BITNET interactive messages, mail (copy message data then execute mail command), and file transfer are the major BITNET services built on the RSCS facility.

• *PhoneNet* provides a store-and-forward electronic mail services that allows CSNET service members to exchange messages with other CREN members and with other major mail networks, including NSFNET, MILNET, etc. Messages are exchanged via a central host at the CSNET Information Center in Cambridge, MA. PhoneNet uses dial-up telephone connections from 1200 to 9600 baud. Messages on PhoneNet, and on the other components of CREN, follow the Internet standard for the format of text messages (RFC 822 and later related documents) [2].

• *Dial-up IP* software allows sites using the switched telephone network to send IP packets through a central server located at the CSNET CIC in Cambridge, Mass, to the Internet. The full DoD Protocol Suite is supported including TCP, UDP, Telnet, FTP, SMTP, rcp, and *rlogin*. Dial-up IP services are based on SLIP (Serial Line IP) and currently run on BSD UNIX platforms only [2].

• *Leased Line IP*—In the past, dedicated IP connections were considered a custom connection service for sites with unusual needs. Today, many CREN members are making use of this style of connection as a common-place method for connecting to CREN, and the Internet.

continued on next page

CREN *(continued)*

CREN manages two dedicated IP "clusters" for members, with one on the East Coast centered at the CSNET CIC, and one on the West Coast, centered at the Olivetti Research Center in Menlo Park, CA. A variety of link speeds are supported up to T1 rates.

• *RSCS over IP*—In addition to CSNET services over IP, the BITNET II protocol system is now available on a limited basis. This protocol package provides an encapsulation of RSCS in IP packets for IBM mainframe machines. The package is under development at Princeton University and has been successfully implemented at six BITNET sites. The initial goal of this protocol is to allow BITNET service hubs to relax the dedicated RSCS BISYNC line in preference to an Internet IP route, if one exists. More information will be available from CREN as this service becomes more widely available.

• *X.25Net* is a CSNET full-service Internet-connected network that uses TCP/IP protocols over the X.25 network Telenet. It provides file transfer, remote login, and immediate electronic mail service between X.25Net hosts [2].

Network services

CREN divides it services into two classes of service: BITNET and CSNET. Members of the original CSNET and BITNET networks rolled over their existing service option into CREN. Note that the services that BITNET and CSNET provide are inherently similar (mail, file transfer, etc) however they are built on very different network technologies. We expect these services to remain distinct for the next several years. (This is not to exclude future advances in interoperability over the next couple years between the CSNET and BITNET services as they become available).

BITNET services

• *Interactive Messaging:* The highest priority traffic in BITNET is the interactive message. Interactive messages are basically a one-line parcel of data that may be sent between users and server (agents) anywhere in the BITNET network. Frequently, these are used to support conversations between people, or communications to a LISTSERV server instructing it to send back a file. The BITNET USERHELP document [3] describes these as "the network's equivalent of telephone conversations."

• *File Transfer:* BITNET provides a way to send data files over the network in a convenient fashion via the SENDFILE facility. This utility is meant to move non-interactive information between people and machines.

• *BITNET Mail:* If you can send a file and then execute a command on the other side, then you've got the basics for building and interactive mail system. BITNET does just that for sending mail between users. (Although the format is different, strong parallels can be drawn to the UUCP manner of sending mail. See *The Matrix* by John S. Quarterman for more information [5]). This is an extensively used facility within the network. BITNET also provides an INTERBIT mail relay service at several points in the network to handle mail exchange between Internet IP networks and the BITNET service network.

• *LISTSERV:* A LISTSERV server is general purpose electronic mail agent that accepts interactive or regular messages and then does something with them. The functions include: sending a message to a distribution list, [un]subscribing to a distribution list, managing a distribution list, requesting that information (e.g., a file) be sent back to the originator, and the like. Electronic mail distribution lists are implemented using LISTSERV servers.

• *NAMESERV:* This is a user directory service designed to help people locate the mail address of intended recipients in the network or to help people locate other people with varied interests. Unfortunately, there are many of these servers and few respond to the same commands or respond in the same manner [3].

CSNET services

CSNET provides basic Telnet, FTP, and SMTP services to all participating IP connected members and electronic mail only for PhoneNet sites. In addition to these basic services, CSNET provides the following:

• *Domain Name Service:* CREN provides its members with primary and secondary domain name services, including coordination with the *Network Information Center* at SRI International, in Menlo Park, CA for the registration and administration of domain names for member sites who do not wish to take on this burden themselves. This has traditionally been a CSNET service, however we expect to be administrating BITNET host MX record information in the domain name service in the near future.

• *CSNET Info-Server:* The CIC offers an electronic-mail based information service via the CSNET "Info Server." This service accepts specially formatted mail from CREN users and automatically returns the requested information by return mail. The Info Server contains a rich set of Internet information topics ranging from address formatting help and ARPA *Request For Comments* documents (RFCs) to the complete `mod.sources` library [2, 4].

• *User Nameserver:* The User Name Server is a database of CSNET users and sites that is maintained on a service host at the CIC. Any authorized CREN users may register in the User Name Server. In addition, every CSNET service member site has an entry containing information about the location, host names, and member representative (liaison) information [2].

• *CSNET Hot-Line:* Staff at the CSNET Coordination and Information Center (CIC) provide technical, operational, administrative, and information services for CSNET service members. The CIC also provides a hotline number which is available twenty-four hours a day, seven days a week [2].

CSNET topology background

Historically, CSNET was built to provide an application level relay service for electronic mail messages exchanged between PhoneNet sites and the ARPA Internet. This meant that CSNET spoke both SMTP, IP, and PhoneNet from the start. Today, we find that the number of PhoneNet sites is reducing in preference for more interactive networking; i.e., DDN services (Telnet, FTP, SMTP) over TCP/IP. We are hoping that many of our PhoneNet sites can roll over to Dial-up IP services. As we move forward into the future, it is reasonable to except that all of the CSNET service hosts will be IP based.

continued on next page

CREN *(continued)*

BITNET topology background

BITNET is built as a very large store-and-forward network in which a host can only exchange information with their neighbor host(s). Until very recently, BITNET service members had to agree to the following extensibility requirement:

"Each Member who participates in BITNET is required to provide at least one port to which another Member or Affiliate may connect to gain connectivity into BITNET..." This formed the foundation principle for the network.

The route from one host to another is based on the information contained in a routing table on that host and there is only one path from a host to any other given host. It has been a policy in BITNET that routes are symmetrical, in that the path between any two hosts traverses the same set of nodes. Routing decisions are straightforward, "to ultimately get to the destination host, I first hand this off to my neighbor host." Routing tables are compiled on a monthly basis (via the NETSERV service) and distributed back out to the network at large. Some sites compile their own routing tables. (This is somewhat similar in concept to how the UUCP network builds its routing information today using *pathalias* [5, 6] with the exception that the majority of nodes in BITNET receive their routing tables from the central NETSERV services rather than build the tables themselves).

In January 1990, the CREN Board or Trustees adopted a new policy that cooperating BITNET nodes, by mutual consent, may substitute an alternate connection technology for the BISYNC line provided that the same full complement of RSCS/NJE services are maintained. This evolution in the fundamental BITNET policy now allows the use of the BITNET II protocol or other solution for connections between BITNET nodes (e.g., IP, DECNet, etc.). This evolutionary step has two goals: 1) link speeds can be substantially increased by substituting an IP path for a 9600 baud BISYNC link, and 2) the requirement to maintain a dedicated BISYNC line has been dropped thereby allowing some sites to make better use of some funding. This change of migrating to RSCS over IP will not change the routing characteristics of the network.

Technical challenges for 1990

CREN is still young and the merger of the technology and services from the CSNET and BITNET worlds will take several, if not many years to complete. For 1990, we have embarked on projects that are more specific to each type of service while keeping an eye on plotting a converging path between two worlds.

CSNET services challenge

The size of the Internet IP world is growing constantly. We are taking active steps aimed at increasing CREN's ability to meet the expanding needs of our membership:

- Providing Dial-IP for new members and as a migration path for existing PhoneNet-only service members.
- Providing CREN-only links between our East and West Coast clusters.
- Registering all CREN members into the Domain Name System.
- Establishing two points of CREN contact to the NSFNET [7, 8, 9] backbone, one via a collaborative effort with NEARNET and one on the West Coast, via one of the NSS sites in California.

- Maintaining our commitment to end-user support.
- Investigating a CREN-only IP backbone network as part of a disaster recover plan.

Our direction in the evolution of CSNET services is to strengthen our presence in the Internet world.

BITNET services challenge

BITNET faces a more interesting evolution along its course for 1990 as it becomes better connected with the CSNET side of the CREN house, and with other networks. Milestones include:

- Establishment of BITNET II services (RSCS/NJE over IP) at as many sites as feasible.
- Publishing the RSCS/NJE in IP protocol as a draft standard.
- Documenting, refining, and widely distributing existing BITNET standards (e.g., requirements for INTERBIT mail relays.)
- Increasing the use of pathalias [6] as a replacement for the existing route building utility GENROUTS.
- Registering BITNET member sites in the Domain Name System.

Our direction in the evolution of BITNET services is to make the network more compatible (interoperable) with the existing services present in the Internet world.

Summary

CREN is unique in its ability to provide network connections and services to a broad range of membership, from small institutions (universities, community colleges, and high schools) to other larger academic and industrial members for the purposes of facilitating the exchange of information consistent with academic, educational, and research purposes. In addition, CREN is an active leader in pursuing cooperative agreements with international academic and research network affiliates. The merger of the CSNET and BITNET technology presents some interesting challenges and we look forward to reporting our successes in future reports.

For more information

CREN Membership:	CREN Office EDUCOM 1112 Sixteenth Street, NW Washington, DC 20036 Phone: 202-872-4200
BITNET Services:	BITNET Network Information Center (BITNIC) EDUCOM 1112 Sixteenth Street, NW Washington, DC 20036 Phone: 202-872-4200 `info%bitnic.bitnet@cunyvm.cuny.edu`
CSNET Services:	CSNET Coordination and Information Center (CSNET-CIC) BBN Systems and Technologies Corp. 10 Moulton Street Cambridge, MA 02138 Phone: 617-873-2777 `cic@sh.cs.net`

continued on next page

CREN *(continued)*

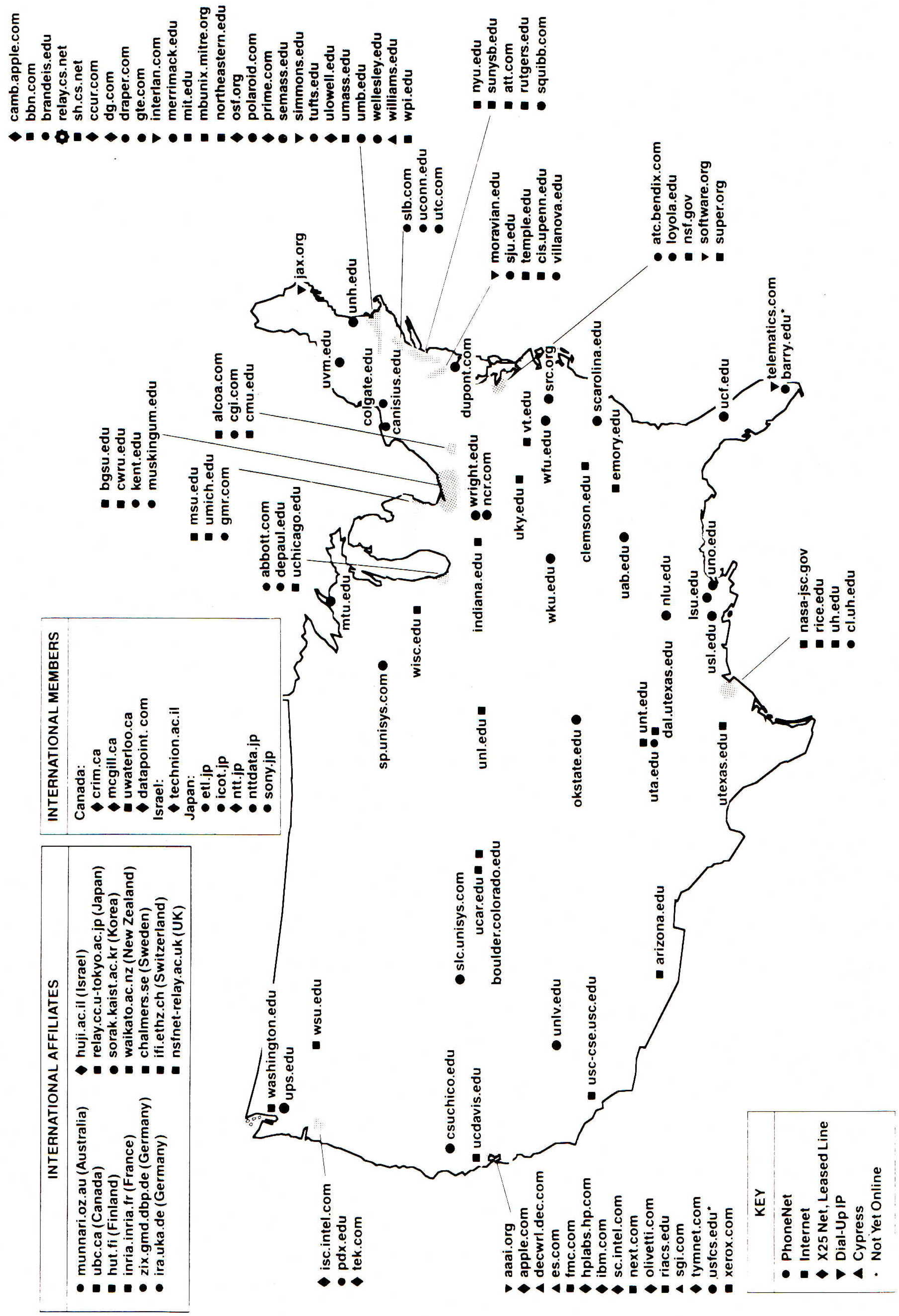

Figure 1: CSNET Geographic Map, August 1989.

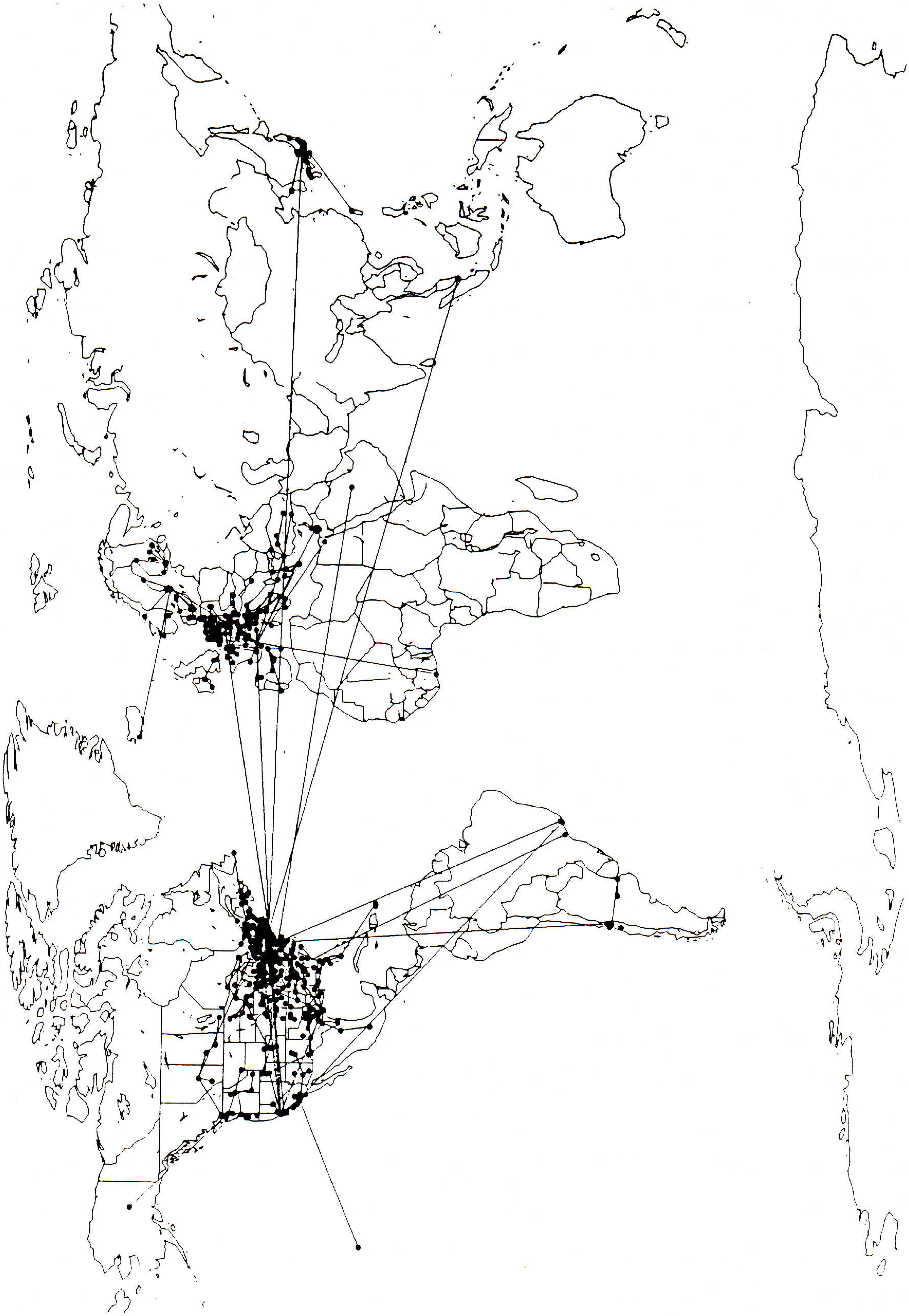

Figure 2: BITNET Geographic Map, September 1989.

continued on next page

CREN *(continued)*

References

[1] The CSNET Coordination and Information Center staff, "The CSNET-FORUM Digest," electronic digest, Volume 5, No. 6, BBN Laboratories Inc., Cambridge, MA, November 1, 1989.

[2] The CSNET Coordination and Information Center staff, "Profile: CSNET—The Computer+Science Network," *ConneXions*, Volume 2, No. 3, March, 1988.

[3] CREN Staff, "The CREN Information Packet," Available from EDUCOM, 1112 Sixteenth Street NW, Washington, DC, 20036, 202-872-4200.

[4] Craig Partridge, Charlotte Mooers, and Mark Laubach, "The CSNET Information Server: Automatic Document Distribution Using Electronic Mail," *Computer Communications Review*, Volume 17, No. 4, October/November 1987.

[5] John S. Quarterman, "The Matrix: Computer Networks and Conferencing Systems Worldwide," Digital Press, 1990, ISBN 1-55558-033-5.

[6] Peter Honeyman and Steven M. Bellovin, "Pathalias or the Care and Feeding of Relative Addresses," Proceedings of the 1986 Summer USENIX Conference (Atlanta, 9–13 June 1986), pp. 368–372, USENIX Association, Berkeley, CA 1984.

[7] NNSC Staff, "Profile: NSFNET," *ConneXions*, Volume 1, No. 2, June 1987.

[8] Hans-Werner Braun, "The new NSFNET backbone network," *ConneXions*, Volume 2, No. 12, December 1988.

[9] Laura Kelleher, "Merit/NSFNET Information Services," *Connexions*, Volume 3, No. 6, June 1989.

MARK LAUBACH received his B.E.E. (1980) and M.S. in Computer Science (1987) from the University of Delaware. For the past ten years, he has worked for the Hewlett-Packard Company on numerous projects starting with Laboratory Automation Systems, continuing recently through electronic mail and networking projects. He was a member of the team that started the internal HP research internet and is currently a member of the HP Internet Review Board and the UNIX Mail Task Force. Currently, he is spending his time researching distributed systems management as a Systems Designer in the Information Architecture Group in Cupertino, CA. From 1988 through 1989, he served on the CSNET Executive Committee. Since September 1989, he has been a member of the Board of Trustees and the Chairman of the Technical Committee for the Corporation for Research and Educational Networking. In addition, he is participating as a member of the Internet Engineering Task Force. Outside of work, Mark and his wife are in the Morgan horse breeding business and he is also on the Board of Directors for the Las Cumbres Amateur Radio Club and is actively participating in Amateur Radio Emergency Services work.

The ACM SIGCOMM Award

The SIGCOMM Award was initiated in 1989 as a means of honoring computer communications professionals for outstanding technical achievement in the fields of data- and computer-communications. The award consists of a plaque and a $2000 honorarium. The award will be presented at the annual SIGCOMM Conference, at which time, the awardee is invited to deliver a technical address.

Guidelines

The following are guidelines for submitting a nomination:

- Self-nominations are not accepted.
- The nominee need *not* be a member of ACM SIGCOMM.
- The nominator *must* be a member of ACM SIGCOMM.
- Nominations must be received by the chairman of the SIGCOMM Award Committee no later than May 30 of each year.
- Nominations which did not result in an award can be resubmitted and updated in subsequent years.
- Previous awardees are not eligible for future nominations. (The 1989 awardee was Paul Baran, for the invention of packet switching).
- Members of the Award Committee are not eligible.
- Members of the SIGCOMM Executive Committee (chairman, vice chairman, secy-treasurer and editor as well as past chairman) are not eligible.

Material to be included in nomination:

- Curriculum Vitae, including publications, of nominee.
- Concise statement of the work for which the award is being nominated. This statement will appear on the award plaque.
- Description of the nominee's role in the work justifying the nomination.
- Letters of recommendation from others which identify rationale for the nomination and by what means the recommender knows of the nominee's work. It is recommended that at least three letters of recommendation be included.
- Justification for declaring the nominee's work to be a major, lifetime contribution to computer communications.

Submitting nominations

The nomination should be made in the form of a letter addressed to the chairman of the SIGCOMM Award Committee. The nominator should solicit recommendations from colleagues in the field who are most familiar with the nominee's achievements. It is not recommended to send copies of published papers or books along with the nomination materials. The nominator is responsible for gathering all nomination materials and sending two copies of all materials to reach the chairman of the awards committee before May 30th. No late nominations will be considered for the current year. They will be held over until the following year for consideration.

The 1990 chair of the SIGCOMM Award Committee is:

Dr. Franklin F. Kuo
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
Tel: 415-859-4116 FAX: 415-859-6171 E-mail: Kuo@nisc.sri.com

Book Review

!%@:: A Directory of Electronic Mail Addressing and Networks, by Donnalyn Frey and Rick Adams. Published O'Reilly and Associates, ISBN 0-937175-93-0, 1989.

If you're looking for a book to lose yourself in, to pick up and not want to put down, then this is *not* the book for you. If you're looking for a book that you'll want to reach for time and time again, one that will help you with your daily job, and one that you'll continuously learn from, then you should invest in *!%@:: A Directory of Electronic Mail* compiled and written by Donnalyn Frey and Rick Adams.

Reference tool

This book is a reference tool that every network pioneer should have for sending electronic mail to those "hard to reach" networks. Now you have no excuse not to send messages to your buddies in Malaysia.

The organization of the book is simple. The bulk of it consists of network descriptions. There are two pages devoted to each network. The first page is a standard layout, which provides a brief, but very useful summary of applications, contacts, addressing syntax, network architecture, future plans, and the date this information was last updated. The second page offers a geographical view of the network, with lines indicating gateways to other networks. Site information, number of hosts and other useful information is given on this page, if known.

E-mail tutorial

In addition to the reference material, there are other very helpful sections. Included is an excellent electronic mail tutorial, written in an easy to understand manner by Daniel Karrenberg and Anke Goos of the *European UNIX systems User Group* (EUUG).

There are also five very handy appendices: second level domains listed by organization and domain name; ISO codes listed by country and code; and an explanation on how Internet addresses are handled by UUCP sites.

Index

A three-way index listing networks by full names (or types), notations, and country of origin, gives you the lookup power you need to keep your mail from bouncing all over the world. There is also a glossary that defines many of the terms used throughout the book.

Update

This is a handbook in the traditional Nutshell style, complete with an animal on the cover (because, as it is stated in the Colophon, "UNIX and its attendant programs can be unruly beasts."). There are hares on this cover, which I personally think is a good choice because it seems like networks these days are multiplying like..., well you know, rabbits! Fortunately this is not a problem because there are plans to update this book regularly. In fact, I am told that a new edition should be available this June.

Get one!

The best aspects of this book are that it is well organized, easy to use and understand, and it will be updated regularly. If you send electronic mail at all, then you must have this book. And while you're at it, please buy one for your site's postmaster! —*Tracy LaQuey*

New Book on Computer Viruses available

Computer Viruses: Dealing with Electronic Vandalism and Programmed Threats, by Eugene Spafford, Kathleen Heaphy, and David Ferbrache, 109 pages. Published by ADAPSO, The Computer and Software Service Industry, 1989.

High-level discussion

The book has been written to be an accessible resource guide for computer users and managers (PC and mainframe). It presents a high-level discussion of computer viruses, explaining how they work, who writes them, and what they do. It is not intended to serve as a technical reference on viruses, both because the audience for such a work would be limited, and because such a reference might serve to aid potential virus authors.

Goal

The goal of the book is to dispel some common myths about viruses (and worms, Trojan Horses, et. al.), and provide simple, effective suggestions for how to protect computer systems against these threats. It furthermore stresses that most systems face greater threats from other areas, so the proper attitude to take is to strengthen overall security; concrete suggestions for enhancing overall security are also presented.

The appendices provide extensive references to other publications, security organizations, anti-viral software sources, applicable (U.S.) state and Federal laws against computer crime, and detailed descriptions of all IBM and Apple Macintosh viruses known as of 1 October 1989.

Audience

Although written for ADAPSO members, almost any computer user should find it instructive. The appendices are an excellent source of further information, addresses and phone numbers, and pointers to software. At least one university professor has indicated he will use the book in a security course, and some law enforcement agencies are also considering using the book for instructional purposes.

Feedback sought

The authors are interested in comments and feedback about the book, especially in areas where information might be added. You can contact them by sending mail to: `virus-book@cs.purdue.edu`.

The book can be ordered from

ADAPSO
1300 North Seventeenth Street
Suite 300
Arlington, VA 22209
USA
Attn.: Mr. John Gracza
Phone: 703-522-5055

[Ed.: We will have a review of this book in an upcoming issue of *ConneXions*, stay tuned!]

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779

Bulk Rate
U.S. POSTAGE
P A I D
SAN JOSE, CA
PERMIT NO. 1